Are Current Prognostic Performance Evaluation Practices Sufficient and Meaningful?

Shankar Sankararaman[1], Abhinav Saxena[2], and Kai Goebel[3]

[1,2] *SGT Inc., NASA Ames Research Center, Moffett Field, CA 94035, USA*
shankar.sankararaman@nasa.gov
abhinav.saxena@nasa.gov

[3] *NASA Ames Research Center, Moffett Field, CA 94035, USA*
kai.goebel@nasa.gov

ABSTRACT

This paper investigates the shortcomings of performance evaluation for prognostic algorithms, particularly in the presence of uncertainty. To that end, the various elements of a prognostic algorithm (present health state estimation, future load condition, degradation model, and damage threshold) and their effects on prognostics are examined. Each of these elements contribute to overall prediction performance and therefore it is important to distinguish between (1) assessment of the correctness of information regarding these quantities, and (2) the assessment of correctness of the prognostic algorithm. The need for proper accounting for uncertainty in the various associated elements is discussed. Next, the shortcomings of traditional comparisons between ground truth and algorithm prediction is discussed. Several scenarios are pointed out where misleading interpretations about evaluation outcomes are possible. In order to address these shortcomings an "informed evaluation" methodology is being proposed, where the algorithm is informed with future loading/operating conditions before comparing against ground truth. Additionally, the importance of estimating the accuracy of aggregating the different sources of uncertainty using rigorous mathematical procedures is also emphasized. While this discussion does not target developing new metrics, it highlights key criteria for an accurate performance evaluation process under uncertainty and proposes new measures to accomplish this goal.

1. INTRODUCTION

1.1. Prognostics

Prognostics, the ability to predict future events, conditional on anticipated usage and environmental conditions, significantly contributes to a system's resilience for safe and efficient operation. It is now well accepted that prognostics can add considerable value to life cycle cost reduction by assessing the state of health of the system components, and estimating their remaining useful life that makes it possible to initiate a mitigating action that will either prevent the breakdown, minimize downtime, avoid unscheduled maintenance, or result in similar outcomes that minimize operational cost of the system. However, at the same time, prognostics is inherently affected by various sources of uncertainty present in the system; if the methods that deal with uncertainty are not adequately understood and incorporated, it can be difficult to make reliable predictions with high accuracy and confidence. It is, therefore, not surprising that considerable attention has been given to this technology in the last few years. A variety of different approaches have been explored and employed to predict system health and/or estimate remaining useful life. However, it is important to note that the term "prognostics" has been used by various practitioners in any context that has a predictive element but not all of these methods result in estimation of remaining life. Subsequently, it also has a bearing on the interpretation and treatment of uncertainty in each of these methods, which is important not only to understand how to incorporate these uncertainties in the analysis but also to assess performance of these methods in a technically correct and rigorous manner (Saxena, Sankararaman, & Goebel, 2014).

1.2. Prognostic Performance Evaluation

Performance assessment of prognostics algorithm is an indispensable element in maturing prognostics and health management technology as these predictions become the basis of any subsequent decision making process. Mitigating actions taken based on these decisions ultimately determine the effectiveness of the overall health management system. Most of the existing literature on prognostics performance evaluation

focuses on choosing the most appropriate metrics to evaluate algorithms. Several metrics have been proposed and used in the past that measure unique characteristics of prognostics (Saxena, Celaya, Saha, Saha, & Goebel, 2010). These metrics described different ways to express and measure accuracy, precision, timeliness, and prediction-confidence attributes of the prediction of a prognostic algorithm. Less attention has been paid towards determining the correct approach for evaluating and interpreting prognostic performance under uncertainty. Current approaches rely on comparing predicted outcomes to observed end of life (also referred to as ground truth). The key question, as investigated in this paper, is whether such a comparison is technically correct, especially when considering uncertainty in the prediction process. In contrast to discussing prognostic metrics, this paper attempts to identify a meaningful approach for performance evaluation irrespective of which metrics are used to quantify performance. In particular, two issues are explored: (1) choosing the baseline to compare prediction results with and (2) identifying a method that can be used to obtain such information. In the process, several important caveats in interpreting the results of prognostic algorithms are explained in detail and several misconceptions are clarified in this regard.

1.3. Relation to Work on Metrics

For providing a clear context with regards to earlier works investigating prognostic performance, it is important to draw connections between the *what should be measured* and *how prognostic metrics were designed*. Early versions of prognostics algorithms output were point estimates of end-of-life that were compared with the observed end-of-life to assess performance (Saxena et al., 2008). Later as prognostics algorithms matured they started incorporating uncertainties in predictions through various representations of uncertainty, although mostly dominated by probability distributions. However, the basic underlying question of what the key contributing factors to the quality of a prediction are and how the contribution of each can be evaluated separately have not been addressed in detail until very recently (Sankararaman & Goebel, 2013b). Prognostic performance is understood to depend on two distinct factors; 1) External inputs (data quality, operating environment, system loading, etc.), and 2) Internal processing (fault models, state estimation methods, uncertainty propagation methods, etc.). To gain full understanding of uncertainty expressed in remaining useful life (RUL) estimates it is important to isolate the effects of these different internal and external factors through adequate performance evaluation while algorithm development. Based on feedback from such evaluation, targets for further technology improvement can be identified and a baseline of acceptable performance can be established before a prognostic system is put into usage. This paper extends the discussion in (Saxena et al., 2014) by focusing on effects of uncertainty in prognostics for the purpose of performance evaluation and explores how carefully designed performance evaluation process can help distill these effects.

1.4. Organization of this Paper

This paper focuses its attention on performance evaluation of only condition based prediction methods for prognostics. Other prediction methods are considered beyond the scope of this paper. First, Section 2 describes various sources of uncertainty that are present in prognostics and clearly distinguishes between the interpretation of uncertainty in condition-based prognostics and fleet-based prediction methods. This discussion dissects the overall uncertainty into a few fundamental elements and subsequently provides a stepwise approach to assess prognostic performance so that these effects of each of these elements on prognostic performance evaluation can be assessed. Next, Section 3 discusses the impact of uncertainty on prognostic algorithms through an illustrative example and a simple prediction algorithm. Section 4 explains the challenges involved in performance evaluation of prognostic algorithms and Section 5 explains different types of performance measures. Section 6 numerically illustrates the above concepts using a lithium-ion battery application. Finally, conclusion and future work are presented in Section 7.

2. PROGNOSTIC ALGORITHMS

In order to completely understand the various aspects of performance evaluation of prognostic algorithms, it is necessary to understand the various elements of a prognostic algorithm. A prognostic algorithm ideally takes all available information (state estimate, future estimates, degradation model, etc.) and computes the remaining useful life of the component or system of interest.

2.1. Key Elements of a Prognostic Algorithm

For the purpose of rating the performance of an algorithm, it is important to decide which elements are part of an algorithm and which are not. Roychoudhury et al. (Roychoudhury, Saxena, Celaya, & Goebel, 2013) focused on identifying the key aspects of a prognostic algorithm, this argument is extended in this paper to identify the various elements that are needed to determine the remaining useful life, as follows:

1. Present condition (state) of the system/component
2. Future (operational, loading, environmental, etc.) conditions of the system/component
3. Degradation model of the system/component
4. End-of-Life damage threshold
5. The actual algorithmic procedure, that combines the above information systematically in order to compute the remaining useful life.

One could argue that quantifying the present condition of the system/component through a state estimation algorithm (perhaps using a Bayesian filtering approach such as particle filtering or Kalman filtering) is a necessary and essential component of the prognostic algorithm. However, the development of the degradation model and estimating the future conditions seem to be outside the scope of the prognostic algorithm. The problem is that these two components are "inputs" to a prognostic algorithm, i.e., the algorithm needs these two pieces of information to predict the remaining useful life. It would not reasonable to penalize an algorithm whose predictions do not compare well with ground truth data, if the algorithm did not have access to an accurate degradation model and/or an accurate estimate of the future conditions of the component/system. Similarly, it is not reasonable to accept a prognostic algorithm whose predictions apparently match well with ground truth data, if the algorithm had used inaccurate future conditions and inaccurate degradation model (whose inaccuracies could cancel each other out). For example, the degradation model may have a much smaller degradation rate and the chosen future conditions may be much more severe than reality.

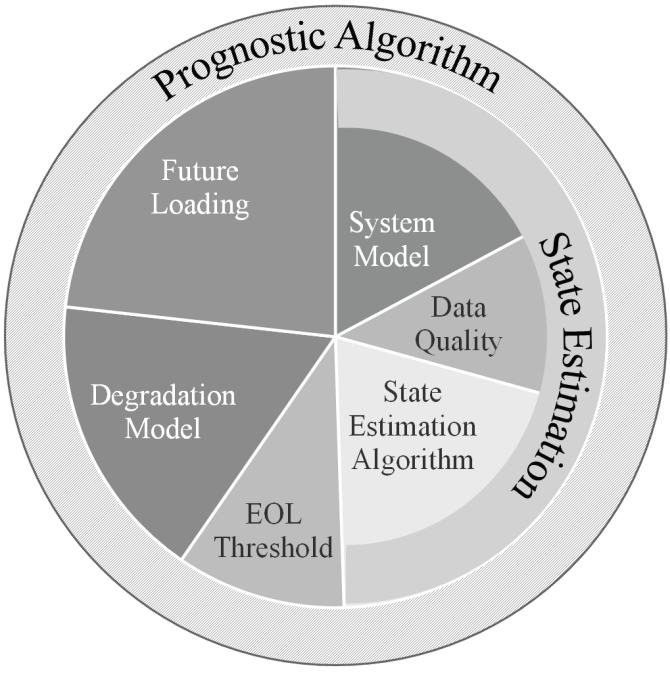

Figure 1. Components of Prognostics Algorithm

Therefore, this paper explores the various aspects of performance evaluation with an emphasis on the above elements of a typical prognostic algorithm, as explained through the rest of this paper.

2.2. Uncertainty in Prognostics

While non-probabilistic methods (Wang, 2011) such as Fuzzy logic, possibility theory, Dempster-Shafer theory, Evidence theory, etc. have been used for the treatment of uncertainty, probabilistic methods have been predominantly used for uncertainty representation in prognostics (DeCastro, 2009; Orchard, Kacprzynski, Goebel, Saha, & Vachtsevanos, 2008; Saha, Goebel, Poll, & Christophersen, 2009). Without loss of generality, the rest of this paper will focus only on prognostic algorithms based on probability theory.

In order to evaluate the performance of prognostic algorithms in the presence of uncertainty, it is important to answer questions such as:

1. What does one actually mean by "uncertainty" in prognostics?
2. What causes uncertainty in prognostics?
3. What are various elements of a prognostic algorithm that are affected by uncertainty?
4. What is the contribution of these elements to overall prognostic performance?

2.3. Interpreting Uncertainty in Prognostics

Though mathematical axioms and theorems of probability have been well-established in the literature and probabilistic methods are being increasingly used for uncertainty quantification in engineering, there is considerable disagreement among researchers on the interpretation of probability. There are two major interpretations based on physical and subjective probabilities, respectively. Physical probabilities (Szabó, 2007), also referred to objective or frequentist probabilities, are related to random physical systems such as rolling dice, tossing coins, roulette wheels, etc. Each trial of the experiment leads to an event (which is a subset of the sample space), and in the long run of repeated trials, each event tends to occur at a persistent rate, and this rate is referred to as the relative frequency". These relative frequencies are expressed and explained in terms of physical probabilities. Thus, physical probabilities are defined only in the context of random experiments. On the other hand, subjective probabilities (De Finetti & de Finetti, 1977) can be assigned to any "statement". It is not necessary that the concerned statement is in regard to an event which is a possible outcome of a random experiment. In fact, subjective probabilities can be assigned even in the absence of random experiments. The Bayesian methodology is based on subjective probabilities, which are simply considered to be degrees of belief and quantify the extent to which the statement is supported by existing knowledge and available evidence. Calvetti and Somersalo (Calvetti & Somersalo, 2007) explain that "randomness" in the context of physical probabilities is equivalent to "lack of information" in the context of subjective probabilities. In this approach, even deterministic quantities can be represented using probability

distributions which reflect the subjective degree of the analyst's belief regarding such quantities.

This leads to the obvious question - is one particular interpretation more suitable to prognostics? In general, both interpretations may be suitable. However, in the particular context of condition-based monitoring or online health monitoring, there is only one system which is being monitored, and hence, at any time instant, there is no "physical randomness" associated with the system (from a frequentist point of view). Therefore, any quantity associated with a system, even though it may be uncertain, cannot be represented using a probability distribution, following the frequentist interpretation of probability. Nevertheless, system state estimation during health monitoring is commonly performed using particle filters and Kalman filters, and these approaches compute probability distributions for the state variables; therefore, the only possible explanation for such calculation is that the subjective (Bayesian) approach is being inherently used for uncertainty quantification. Such filtering approaches are known as "Bayesian tracking" methods not only because they make use of Bayes theorem, but also fall within the realm of subjective probability. This implies that the uncertainty estimated through the aforementioned filtering algorithms are simply reflective of the analyst's degree of belief, and not related to actual physical probabilities. Similarly, the uncertainty in future conditions (loading, operating, and environmental conditions) also need to interpreted subjectively. For example, if the anticipated current on a battery follows a normal distribution with mean and standard deviation equal to 10 and 1 (current units) respectively, then this probability distribution is only reflective of the subjective belief, and only one realization may occur in reality. The actual current may be 10 units (which is not possible to know), and this implies that the subjective belief was reasonable; the subjective belief would have been even better had the standard deviation been smaller. On the other hand if the actual current had been 30 units, then it implies that the subjective belief was completely wrong.

Sometimes, in practice, both frequentist and subjective information can be useful, even in condition-based prognostics. For example, an ensemble of test units may be used to develop degradation models and learn the corresponding model parameters. Since these models and their parameters are estimated based on physically variable units, the uncertainty in such parameters need to be interpreted from a frequentist point of view. However, when such a model is used in condition-based monitoring, these parameters are typically updated in order to reflect the parameters of the particular unit; during this procedure, the interpretation of uncertainty transitions from "frequentist" to "subjective" as the information described in terms of uncertainty changes from reflecting the ensemble of test units to the particular unit under consideration for condition-based monitoring. It is important to understand the interpretation of uncertainty during the course of the monitoring procedure, depending upon what information is used to characterize and quantify the aforementioned uncertainty.

2.4. Sources of Uncertainty in Prognostics

Having discussed the importance and interpretation of uncertainty, this subsection seeks the answer to the question: What are the different sources of uncertainty in prognostics? Typically, the answer to this question varies from application to application, and depends on the type of prediction. For example, in testing-based prediction methods (referred to as "reliability-based testing" in some publications), the remaining useful life is typically calculated by testing multiple nominally identical specimens of the engineering component/system. It may be noted that the term "remaining" in "remaining useful life" may not be applicable to such testing methods. This is because, testing is typically carried out before the engineering system is under operation. The term "time-to-failure" is more appropriate for testing-based health management. It is important not to confound "time-to-failure" and "remaining useful life".

Assume that a set of run to failure experiments have been performed with high level of control, ensuring same usage and operating conditions. The time to failure for all the n samples ($r_i; i = 1$ to n) are measured. It is important to understand that *different* time-to-failure values are obtained due to inherent variability across the n different specimens, thereby confirming the presence of physical probabilities or true randomness. The various factors that contribute are:

1. Inherent variability in properties and characteristics of the nominally identical specimens
2. Inherent variability across the loading conditions experienced by each of the individual specimens
3. Inherent variability in operating and environmental conditions for each of the individual specimens

On the other hand, in condition-based prognostics, the focus should be on monitoring the performance of one particular component/system where the inherent variability across nominally identical units are not of interest. In other words the end of life of the system under test is not governed by system to system variability within the context of condition based predictions or prognostics. It is, therefore, necessary to adopt a significantly different approach for the treatment of uncertainty. Various uncertainties involved in prognostics can be divided into following broad categories:

1. **Present uncertainty:** Prior to prognosis, it is important to be able to precisely estimate the condition/state of the component/system at the time at which RUL needs to be predicted. Typically, damage (or faults) are expressed in terms of states, and therefore, estimating the state is equivalent to estimating the extent of damage (or fault).

This is related to state estimation and is commonly addressed using filtering. Output data (usually collected through sensors) are used to estimate the state and many filtering approaches (Kalman filtering, particle filtering, etc.) are able to provide an estimate of the uncertainty in the state. Practically, it is possible to improve the estimate of the states and thereby reduce this uncertainty, by using better sensors and improved filtering approaches. It is important to understand that the system is at a particular state at any time instant, and the aforementioned uncertainty simply describes the lack of knowledge regarding the "true" state of the system.

2. **Future uncertainty:** The most important source of uncertainty in the context of prognostics is due to the fact that the future is unknown, i.e. the loading, operating, environmental, and usage conditions are not known precisely, and it is important to assess this uncertainty before performing prognosis. If there is no uncertainty regarding the future, then there would be no uncertainty regarding the *true* remaining useful life of the engineering component/system. However, this true RUL needs to be estimated using a model; the usage of a model imparts additional uncertainty as explained below.
3. **Modeling uncertainty:** It is necessary to use a functional degradation model in order to predict future state behavior, i.e., model the response of the system to anticipated loading, environmental, operational, and usage conditions. Further, the end-of-life is also defined using a Boolean threshold functional model, that is used to indicate whether failure has occurred or not. These two models are jointly used to predict the RUL, and they may either be physics-based or data-driven. It may be practically impossible to develop models that accurately predict the underlying reality. Modeling uncertainty represents the difference between the predicted response and the true response (that can neither be known nor measured accurately), and comprises of several parts: model form, model parameters, and process noise. While it may be possible to quantify these terms until the time of prediction, it is challenging to know their values at future time instants.

3. IMPACT OF UNCERTAINTY ON PROGNOSTIC ALGORITHMS

To better illustrate the impact of uncertainty on prognostic algorithms, a conceptual example is introduced in this section.

3.1. Conceptual Example

Consider an engineering component whose health state at any time instant is given by $x(t)$. Consider a simple degradation model, where the rate of degradation of the health state (that decreases with time, due to the presence of damage) is proportional to the current health state. This can be mathematically expressed as:

$$\dot{x}(t) \propto x(t), \tag{1}$$

where the constant of proportionality is a negative number. Since differential equations are usually solved by considering discrete time instants, the above equation can be rewritten as:

$$x(k+1) = a.x(k) + b, \tag{2}$$

where k represents the discretized time-index. The condition that "the constant of proportionality in Eq. 1 is negative" is equivalent to the condition that "$a < 1$ in Eq. 2". The initial health state, i.e., $x(0)$ is a random variable, and is expressed using a probability distribution. For the sake of illustration, let a denote the loading on the system (the smaller the value a, the larger the degradation rate), and let b denote the parameter of the above degradation model. While a and b are constant and time-invariant (for the sake of illustrating the conceptual example), they are random and expressed using probability distributions. (In practical examples, the probability distributions of a and b could vary as a function of time.)

In order to compute the remaining useful life, it is necessary to chose a threshold function that defines the occurrence of failure. Since $x(k)$ is a decreasing function, the threshold function will indicate that failure occurs when the state value x becomes smaller than a critical lower bound (l), and the first time instant at which this event occurs indicates the end of life, and this time instant can be used to calculate the RUL. For the purpose of illustration, consider prediction at the initial time instant; hence, the end of life is equal to the remaining useful life. This remaining useful life (r, an instance of the random variable R) is equal to the smallest n such that $x(n) < l$, and is expressed as:

$$r = \inf\{n : x(n) < l\}, \tag{3}$$

In general (i.e., at arbitrary time instants when it is desired to make prediction), the RUL is calculated as the difference between the end-of-life and the time of prediction.

3.2. Closed-Form Solutions?

This section postulates that closed-form analytical solutions for the remaining useful life prediction are not available even for such simple problems involving linear prediction models. In order to illustrate this point, assume that the chosen time-discretization level is infinitesimally small, it is possible to directly estimate the RUL by solving the equation:

$$a^r.x(0) + \sum_{j=0}^{j=r-1} a^j.b = l. \tag{4}$$

The above equation can be used to calculate the RUL (r) as a function of the initial state ($x(0)$), loading (a) and model parameter (b). For the sake of further simplification, assume

that a and b are completely known constants and $x(0)$ is the only uncertain quantity; further assume that $x(0)$ follows a Gaussian distribution. The following analysis shows that it is impossible to analytically calculate the remaining useful life prediction even with only one uncertain variable and a linear degradation model.

The RUL R follows a Gaussian distribution if and only if it is linearly dependent on $x(0)$. In other words, R follows a Gaussian distribution if and only if Eq. 4 can be rewritten as:

$$\alpha.r + \beta.x(0) + \gamma = 0 \tag{5}$$

for some arbitrary values of α, β, and γ. If it were possible to estimate such values for α, β, and γ, the distribution of RUL can be obtained analytically.

In order to examine if this is possible, rewrite Eq. 4 as:

$$x(0) = \frac{1}{a^r}(l - \sum_{j=0}^{j=r-1} a^j.b) \tag{6}$$

While $x(0)$ is completely on the left hand side of this equation, r appears not only as an exponent in the denominator but is also indicative of the number of terms in the summation on the right hand side of the above equation. Therefore, it is clear that the relationship between r and $x(0)$ is not linear. Therefore, even if the state variable ($x(0)$) follows a Gaussian distribution, the RUL (r, a realization of R) does not follow a Gaussian distribution. Thus, it is clear that even for a simple problem consisting of linear state models, a straightforward threshold function, and only one uncertain variable that is Gaussian, the calculation of the probability distribution of R is not trivial. Even the distribution type of RUL is unknown for this conceptual problem.

Indeed practical problems considered in the prognostics and health management domain may consist of:

1. Several non-Gaussian random variables which affect the RUL prediction,
2. A non-linear multi-dimensional state space model,
3. Uncertain future loading conditions,
4. A complicated threshold function which may be defined in multi-dimensional space.

It is the goal of a prognostic algorithm to rigorously account for all the uncertain quantities and compute the uncertainty in the remaining useful life prediction. It is important to note that RUL is simply a dependent quantity and needs to be predicted without making any assumptions regarding the distribution type (say, Gaussian) or statistics (say, mean or standard deviation) of RUL. This can be addressed posing RUL prediction as an uncertainty propagation problem (Sankararaman & Goebel, 2013b, 2013a). For this purpose, the remaining useful life prediction needs to be written as a function of all of the uncertain quantities. For instance, in the above conceptual example, Eq. 4 can be rewritten as:

$$r = G(x(0), a, b) \tag{7}$$

Then, the uncertainty in $x(0)$, a and b are propagated through G (note that G is equivalent to solving Eq. 4 for r) to compute the uncertainty in the remaining useful life prediction. In the case of practical problems, such computation is very challenging particular when prognostic calculations need to be performed during the operation of the system.

3.3. Conceptual Algorithm

Given information regarding the state estimate, future conditions, and degradation model, this section further uses a conceptual algorithm for the purpose of illustration. This algorithm calculates the mean and standard deviation of RUL using first order Taylor's series expansion (Sankararaman, Daigle, & Goebel, 2014), and is known as the first-order second moment (FOSM). Note that this simply has been deliberately chosen to illustrate certain pitfalls of existing performance evaluation methods.

For the conceptual example of Section 3.1,

$$\mu_r = G(\mu_{x(0)}, \mu_a, \mu_b) \tag{8}$$

where μ_r, $\mu_{x(0)}$, μ_a, μ_b denote the mean of r, $x(0)$, a, and b respectively. The variance of r, i.e., σ_r^2 can be calculated as:

$$\sigma_r^2 = (\frac{\partial G}{\partial x(0)})^2\sigma_{x(0)}^2 + (\frac{\partial G}{\partial a})^2\sigma_a^2 + (\frac{\partial G}{\partial b})^2\sigma_b^2 \tag{9}$$

where σ_r, $\sigma_{x(0)}$, σ_a, σ_b denote the standard deviation of r, $x(0)$, a, and b respectively.

Typically, $\mu_{x(0)}$ and $\sigma_{x(0)}$ are provided by the state estimation algorithm, and the RUL needs to be predicted by forecasting (extrapolating using the degradation model) the state estimate forward in time; such forecasting is equivalent to the calculation in Eq. 7. For example, consider the following statistics: $x(0)$ follows a Gaussian distribution (with mean and standard deviation equal to 1000 and 200 respectively), a follows a uniform distribution (with lower and upper bounds of 0.990 and 0.995), and b follows a uniform distribution (with lower and upper bounds of -0.005 and 0 respectively). For failure threshold limit $l = 50$, the RUL prediction can be approximated to be a Gaussian distribution based on the above calculation of the FOSM method. The resultant probability density function (PDF) is indicated in Fig. 2.

The various aspects of performance evaluation are discussed in detail using this algorithm. While the above algorithm is simply used for the purpose of illustration, the following discussion can be extended to any type of unit-based prognostic algorithm.

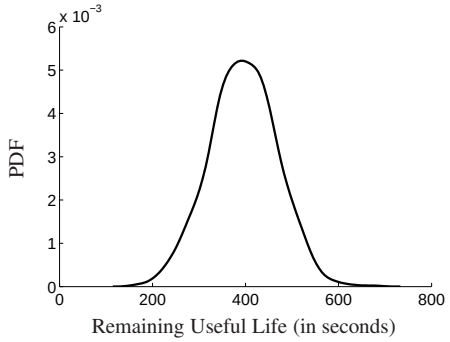

Figure 2. RUL: Conceptual Example

4. CHALLENGES IN PERFORMANCE EVALUATION

Any typical prognostic algorithm uses information regarding the three key elements, i.e., state uncertainty, future uncertainty, and model uncertainty, and computes the remaining useful life prediction. While it would be ideal to compute the entire probability distribution of the RUL, some algorithms compute only certain statistics (like mean and standard deviation) and assume a distribution type (such as Gaussian). Recall that Section 3 stipulated that such assumptions should not be made, and RUL must be fully treated as a dependent quantity.

In order to judge the performance of an algorithm, ground truth data are obtained through experimental studies that *mimic* the various uncertainties that are accounted for, in the prognostic algorithm. Note that it is not individually possible to evaluate how well each of the three key elements have been quantified; only their combined effect on the RUL prediction can be compared against ground truth data.

As far as experiment is concerned, the component/system is at a particular state at any instant of time and there is no uncertainty regarding this state. However, a typical state estimation cannot precisely estimate this state and hence, expresses the uncertainty through a probability distribution. Hence, a typical state estimation algorithm adds extraneous uncertainty, and this would not exist if an idealistic state estimator were present. Similarly, the degradation model uncertainty is also extraneous from the perspective of an algorithm (arises due to the inability to accurately predict the underlying degradation phenomenon), and would not exist if an idealistic, exact degradation model were used. These two types of uncertainty cannot be simulated in a laboratory experiment since they are extraneously added by the algorithm due to the lack of an exact state estimate and an exact degradation model. In fact, effect of state estimation uncertainty and model uncertainty on the difference between the the ground truth and prediction will be equal to zero in the presence of an exact state estimate and an exact degradation model.

However, this is not the case for future loading uncertainty because this uncertainty represents possible future realizations of loading conditions. Hence, it is possible to simulate multiple future loading conditions in the laboratory. However, the challenge lies in the fact that one unit can experience only one set of loading conditions. Multiple loading conditions would have to be simulated on multiple, nominally identical units, and in this case, run-to-failure times of these multiple, nominally identical units will be colored by the inherent variability across them. Hence, it is not possible to experimentally emulate multiple future loading conditions, in the context of condition-based monitoring. And, it is not possible to rigorously evaluate prognostic algorithm performance by considering the simultaneous, joint, effect of state estimation uncertainty, model uncertainty, and future uncertainty on the remaining useful life prediction. Therefore, it is necessary to investigate other practical performance evaluation techniques that can quantitatively judge quality of the remaining useful life predictions of a prognostic algorithm.

5. PRACTICAL PERFORMANCE EVALUATION

This section discusses the most common method of performance evaluation, i.e., comparing the actual run-to-failure time against the algorithm prediction. The shortcomings of this approach are described and new performance evaluation approaches are suggested.

5.1. Ground Truth Comparison

Most existing performance evaluation techniques rely on the availability of the ground truth failure data, and the RUL predicted by the prognostic algorithm can be easily compared against the observed failure time. However, such comparison is not only inequitable, but,sometimes, it may lead to incorrect conclusions.

1. **Inequitable Comparison:** From the time of prediction until the time of failure, the algorithm assumes some uncertainty regarding the future loading and usage conditions. However, the observed ground truth is reflective of only one loading/usage condition that actually happened in reality, thereby implying that similar quantities are not compared. In other words, the experiment contains no uncertainty regarding loading/operating conditions, whereas the algorithm accounted for such uncertainty.
2. **Concluding poor performance of a good algorithm:** The aforementioned inequitable comparison can sometimes lead to concluding that a good algorithm is poor. Consider the case where an algorithm is provided future loading conditions that are completely different from the actual loading conditions. The algorithm may process the provided information accurately and compute the RUL. However, this prediction may be completely different from the observed ground truth RUL. This difference needs to be attributed only to the incorrectly as-

sumed loading conditions and it is not reasonable to penalize the prognostic algorithm in this context. In the context of the conceptual example, the actual loading may have been corresponding to $a = 0.90$ which would have led to a much smaller ground truth RUL than that predicted by the algorithm in Fig. 2. Thus, though the algorithm had been reasonably accurate, its performance would have been judged based on incorrect loading assumptions.

3. **Concluding good performance of a poor algorithm:** Suppose that the prediction of the algorithm is extremely accurate and precise, with respect to the observed ground truth. Then, it cannot be inferred that the algorithm is performing well. This is because the algorithm may not be accurately processing all the uncertainty regarding the future and thereby leading estimates with lesser precision than what the algorithm is supposed to do.

Some of these challenges can be overcome using another type of performance evaluation, as explained in the following section.

5.2. Informed Ground Truth Comparison

It is possible to eliminate the effect of not knowing the loading condition in advance, by waiting until failure. The actual loading/usage condition experienced by the component/system can be observed, and the prediction algorithm can be provided this information. Therefore, the algorithm prediction can be "informed" with the actual loading condition, and the informed-prediction can be computed easily. Note that, at the time of prediction, this information would generally not be available to the algorithm. Therefore, this procedure is only to evaluate the algorithm performance, after eliminating the effect of unknown future loading conditions. All the other information provided to the algorithm need to be reflective of the information available to the algorithm at the time of prediction, such as the state values at the instant of prediction.

In the conceptual example, assume that a component has been run until failure, and the actual loading condition was observed to correspond to $a = 0.994$. Then, the informed prediction can be computed, as shown in Fig. 3. Note that the original prediction has also been shown, for the sake of comparison. This comparison needs to confirm that the observed ground truth falls within reasonable bounds of the informed prediction; note that these bounds are much narrower than the bounds corresponding to the original algorithm prediction.

Similar to the traditional ground-truth-based evaluation, the informed prediction of the algorithm can be compared against the observed ground truth. Note that the former is uncertain because of uncertainty in the state estimate and the degradation model. Note that it is still difficult to evaluate the effects

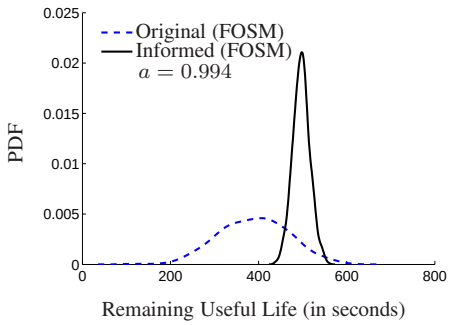

Figure 3. RUL Prediction: Original vs. Informed

of state estimation uncertainty and model uncertainty; in fact, these two quantities could have compounding or canceling effects and such effects cannot be detected and evaluated easily, unless intermediate measurements of the state are available during the experimental set up.

5.3. Assessment of Computational Accuracy

While the above described measures of evaluation focus on characterizing the effects of state estimates, future loading conditions, and degradation model, it is also necessary to check whether the algorithm is accurately processing the different sources of uncertainty. This is not related to accurately predicting the RUL, but is directly associated to the mathematical treatment of the various sources of uncertainty. Some algorithms may average the effect of the different sources of uncertainty on the RUL, and arbitrarily calculate the variance of RUL using approximations and assumptions (Sankararaman & Goebel, 2013b). It is important not to underestimate or overestimate the underlying uncertainty and accurately calculate the probability distribution of RUL. The ideal approach to perform such calculation is the use of Monte Carlo simulation with a large number of samples; though this requires high computational power, this method can be used to check the performance of other algorithms that are suitable for online prediction. In other words, the probability distributions obtained using the specific algorithm and Monte Carlo simulation can be compared and any discrepancy can be quantified, in order to evaluate the performance of the algorithm, from the perspective of integrating the different sources of uncertainty.

For instance, in the conceptual example, if $x(0)$ follows a Gaussian distribution (with mean and standard deviation equal to 1000 and 200 respectively), a follows a uniform distribution (with lower and upper bounds of 0.990 and 0.995), and b follows a uniform distribution (with lower and upper bounds of -0.005 and 0 respectively), then the RUL (defined by Eq. 3, where $l = 50$) can calculated as a probability distribution, using Monte Carlo sampling. Using unit discretization (i.e., the time interval between the k^{th} and $(k+1)^{th}$

instants is equal to one second) for solution, the resultant probability density function (PDF) obtained using exhaustive Monte Carlo sampling (MCS) is shown in Fig. 4. For the sake of comparison, the previously obtained result using FOSM is also shown.

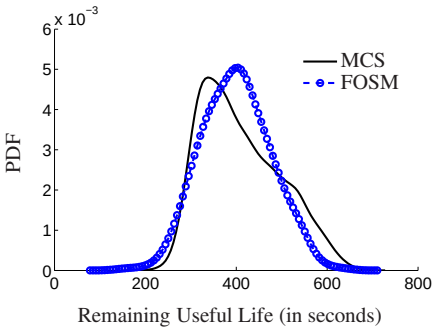

Figure 4. RUL: Conceptual Example

An ideal algorithm should be able to replicate the result from Monte Carlo sampling, as much as possible. A narrower prediction implies that the algorithm is underestimating the total amount of uncertainty whereas a wider prediction implies that the algorithm is overestimating the total amount of uncertainty. The former scenario may lead to unexpected system failure and hence heavy losses, whereas the latter scenario results in extremely conservative decisions and may not use the available resources in an optimal manner.

Note that the FOSM method reasonably agrees with MCS, in this example. This can be attributed to the fact that the the example itself was very simple to begin with. When more uncertain variables are present, and when the degradation model becomes increasingly non-linear, then it is expected that the FOSM result will be significantly different from the MCS result.

5.4. Summary

The search of prognostic performance evaluation measures raises several important questions and concerns. There are four important critical factors that control the performance of prognostic algorithm, and it is not practically possible to individually evaluate the goodness of these factors. While evaluating algorithm performance against observed ground truth seems to be the most widely used method, it is not only unfair but may lead to incorrect conclusions. The informed-prediction method eliminates the uncertainty regarding the future loading conditions, and quantifies the combined effect of state uncertainty and degradation model uncertainty on the RUL prediction. The fourth factor, i.e., whether all the sources of uncertainty are being processed and integrated accurately, can be verified by comparing the algorithm prediction against rigorous Monte Carlo simulation.

An important challenge is the inability to check whether the loading conditions assumed by the algorithm are reflective of what is expected in reality. Is it reasonable to penalize the algorithm for poor performance? Another issue is the ability to identify whether the adverse effect of two (or more) incorrectly estimated quantities jointly cancel out one another, and deceivingly suggest that the prediction is highly accurate and precise. Further research is necessary to address these issues and advance the state-of-the-art in performance evaluation of prognostic algorithms.

6. AN ILLUSTRATIVE EXAMPLE

This section provides an application example to illustrate the various concepts explained earlier in this paper. The example used in this paper predicts end-of-discharge of a Li-ion battery and is borrowed from previous works of the authors (Sankararaman et al., 2014). Since various details about prognostic model development for Li-ion battery are not directly relevant to this discussion they are omitted here, which can be found in (Sankararaman et al., 2014). This example illustrates how one can apply the evaluation method proposed in Section 5 to a real problem. To illustrate pitfalls of raw ground truth comparison and explain the proposed methodology, the rest of this section discusses the various sources of uncertainty in this application example, and explains the previously discussed performance measures.

6.1. Sources of Uncertainty

Consider the prediction of end-of-discharge (EOD) at the initial time instant (t_0). The EOD prediction depends on the following uncertain quantities:

1. **State Uncertainty**: Typically, state estimation is addressed using a filtering technique that can continuously estimate the uncertainty in the state based on the available measurements. In the example discussed in (Sankararaman et al., 2014; Daigle, Saxena, & Goebel, 2012) there are three state variables tracking amount of charge in three capacitive elements of the battery model. These three capacitive elements are referred to as — bulk capacitance (C_b); concentration-polarization capacitance (C_{sp}); and ohmic-drop capacitance (C_s). For complete details of the battery model, and explanation of these terms, refer to (Sankararaman et al., 2014; Daigle et al., 2012).
   It must be noted that in this problem, the charge in C_b is the most influential state variable for predicting the end-of-discharge, and therefore, is considered to be the only uncertain state variable. At the initial time instant, the value of the state variable C_b is denoted by X, and the values of the other state variables are set to zero. Let μ_X and σ_X denote the mean and standard deviation of X.
2. **Loading Uncertainty**: For the purpose of illustration and simplicity, the future loading is assumed to be con-

stant; however, this constant value is chosen at random, and denoted by Y. Let μ_Y and σ_Y denote the mean and standard deviation of Y.

All other quantities are assumed to be completely known constants. The above two sources of uncertainty are sufficient to explain the concepts discussed in this paper.

6.1.1. End-of-Discharge Prediction and Performance Evaluation

It can be seen that the end-of-discharge (EOD) can be written as a function of the uncertain quantities (X and Y), as:

$$EOD = G(X, Y) \tag{10}$$

Note that G is a combination of the degradation model and the end-of-discharge voltage threshold (V_{EOD}) mentioned earlier, and includes all constants that are precisely known. Due to the uncertainty in X and Y, the predicted EOD is also uncertain and represented using a probability distribution. This distribution needs to be compared against experimental end-of-discharge data for performance evaluation. The remainder of this section illustrates various aspects of prognostic algorithm performance evaluation under uncertainty.

6.2. Rejecting a Good Algorithm

If prognostics and prognostics performance are not interpreted and understood correctly, then it may lead to inferring that the algorithm is not performing well.

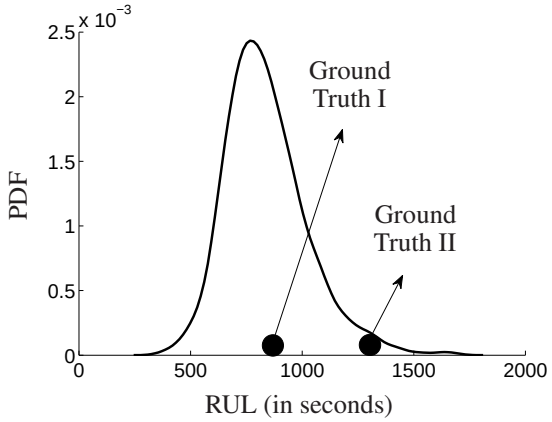

Figure 5. Rejecting a Good Algorithm

For example, consider the RUL prediction (equal to the end of discharge, since the prediction is performed at $t = 0$) in Fig. 5, obtained through Monte Carlo sampling. In this illustration, X and Y are chosen to be Gaussian variables, with $\mu_X = 31115.0$, $\sigma_X = 3111.5$, $\mu_Y = 35$, and $\sigma_Y = 5$. In addition to the RUL prediction, two different ground truth RUL values (Ground Truth I and II respectively) are shown; these two values correspond to different future loading realizations – the more severe results in a shorter life whereas the less severe results in a longer life.

Evidently, the comparison suggests that the algorithm is not performing well since it does not predict Ground Truth II well. However, this may have happened due to several reasons such as:

1. Overestimating the system health during state estimation that leads to the early prediction

2. Overestimating the severity of the loads that leads to early prediction

There is nothing wrong about the algorithm; the information provided to the algorithm is alone questionable. Further, note that the above comparison against the ground truth is unfair since the ground truth represents only one out of several possible realizations considered in the prognostic algorithm.

6.3. Accepting a Bad Algorithm

On the other hand, consider an algorithm that produces the RUL prediction as shown in Fig. 6, and assume that Ground Truth II alone was available through experiments. For example, such an algorithm may compute the RUL in a completely wrong approach in predicting the RUL either by neglecting certain sources of uncertainty or by incorrectly combining the state information along with the degradation model and the threshold model. Therefore, this may lead to concluding that the algorithm is performing well.

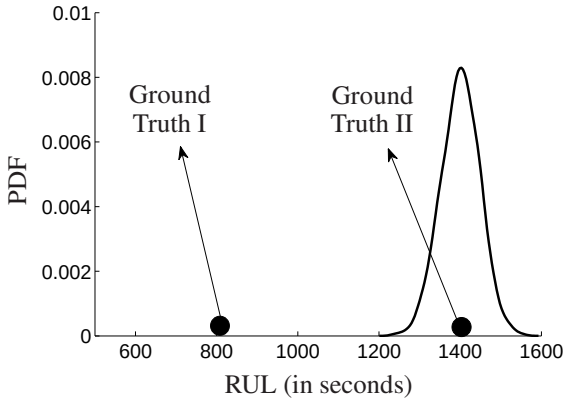

Figure 6. Accepting a Bad Algorithm

However, such a conclusion is incorrect. Since some uncertainty is not accounted for, this algorithm can only capture certain possible realizations of the future but not all possible future realizations; in this case, while Ground Truth II alone be explained by the algorithm, Ground Truth I (which is also a possible future realization) cannot be explained by the algorithm.

6.4. Performance Evaluation

In order to address these issues, this paper discussed two additional measures for performance evaluation. For the purpose of illustration, assume that the FOSM algorithm has been pursued. The first measure of "informed" evaluation measures the actual loading scenario (value of Y, the electrical current, in this numerical example) experienced by the ground truth and "informs" the algorithm with such ground truth. In this case, $Y = 35$ corresponds to Ground Truth I, $Y = 25$ corresponds to Ground Truth II, i.e., a less severe loading leads to longer life. The informed predictions are plotted in Fig. 7, and it can be easily seen that both informed RUL predictions match well with the corresponding ground truth values.

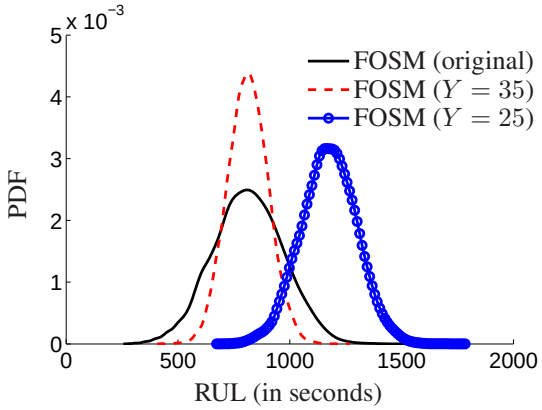

Figure 7. FOSM: Original vs. Informed

The second measure focuses on evaluating the correctness of the algorithm by direct comparison against rigorous Monte Carlo simulation, as shown in Fig. 8. As it can be seen from this figure, the FOSM algorithm is able to capture central tendencies but is not able to capture tail behavior. For this numerical example, the prediction seems to be conservative. However, it could be otherwise for a different set of uncertain quantities and corresponding statistics. That is why it is important to evaluate such correctness by direct comparison against MCS.

6.5. Discussion

Practical problems may have several sources of uncertainty that further complicate performance evaluation through complicated interactions, i.e., Eq. 10 may get complicated with multiple arguments. Many of these sources of uncertainty are "inputs" to the prognostic algorithm, and it is not reasonable to penalize the algorithm if the information regarding these "inputs" are incorrect. That is why it is necessary to develop a rigorous approach to separate (1) evaluation of correctness of information regarding these "inputs" from (2) evaluation of the prognostic algorithm itself. This paper presented a few preliminary steps in this direction and future research may continue to explore the topic of prognostic performance evaluation in further detail.

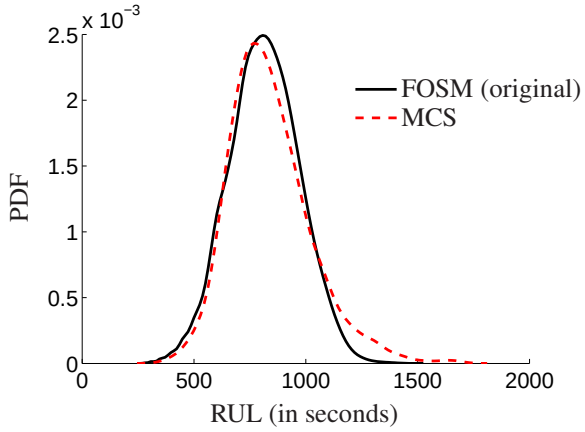

Figure 8. FOSM Algorithm vs. MCS

7. CONCLUSION

This paper discussed the various aspects of performance evaluation of prognostic algorithms in detail, particularly in the presence of uncertainty. To begin with, it was explained that there are several sources of uncertainty that affect prognostics, and that a good prognostic algorithm needs to rigorously account for all of these uncertainties and quantify their combined effect on the remaining useful life prediction. While the presence of uncertainty has been addressed using probability methods, it was explained that the interpretation of probability is not straightforward in prognostics. In testing-based prediction methods, there is inherent variability amongst all the nominally identical specimens that are being tested, and classical statistics-based or frequentist interpretation is applicable. However, in condition-based monitoring, only one unit is studied; therefore, physical variability is absent and all uncertainty needs to be interpreted subjectively. This difference in interpretation plays a key role in understanding the various elements that effectively contribute to the performance of a prognostic algorithm. These elements include: (1) state estimate and associated uncertainty; (2) future loading, operating, and environmental conditions, and associated uncertainty; (3) degradation model and associated uncertainty; and (4) end-of-life threshold and the associated uncertainty. Then, this paper discussed methods for performance evaluation from the perspective of quantifying the combined effect of these elements on the remaining useful life prediction.

First, this paper postulated that it is not possible to evaluate algorithm performance by simultaneously accounting for all these three sources of uncertainty. Second, the most popular technique of comparing ground truth against the algorithm prediction was discussed, and its shortcomings were mentioned. This approach is not only unfair, but also may lead to incorrect conclusions of rejecting a correct algorithm and accepting a wrong algorithm. In order to address some short-

comings of this approach, an "informed evaluation" methodology was proposed; in this method, the true future loading information (available after failure) is provided to the algorithm and then, it is tested whether the ground truth falls within reasonable bounds of the algorithm prediction. Finally, the importance of the mathematical treatment of the different sources of uncertainty was explained, and in this context, it is necessary to compare the performance of any algorithm against Monte Carlo simulation. In other words, given the same information to the algorithm and Monte Carlo simulation, the algorithm prediction needs to be "similar" (in fact, as exact as possible) to that of the Monte Carlo prediction. A narrower prediction implies that the algorithm is underestimating the total amount of uncertainty whereas a wider prediction implies that the algorithm is overestimating the total amount of uncertainty. Future work needs to further explore the concepts of informed evaluation and identify metrics that can express various performance aspects of a prognostic algorithm.

ACKNOWLEDGMENT

The work reported herein was in part funded by the NASA System-wide Safety Assurance Technologies (SSAT) project under the Aviation Safety (AvSafe) Program of the Aeronautics Research Mission Directorate (ARMD) and by the AGSM (Advanced Ground Systems Maintenance) project under the Ground Systems Development and Operations Program in the Human Exploration and Operations Mission Directorate.

REFERENCES

Calvetti, D., & Somersalo, E. (2007). *Introduction to Bayesian scientific computing: ten lectures on subjective computing* (Vol. 2). Springer.

Daigle, M., Saxena, A., & Goebel, K. (2012). An efficient deterministic approach to model-based prediction uncertainty estimation. In *Annual conference of the prognostics and health management society.*

DeCastro, J. A. (2009). Exact nonlinear filtering and prediction in process model-based prognostics. In *Annual conference of the prognostics and health management society.* San Diego, CA..

De Finetti, B., & de Finetti, B. (1977). Theory of probability, volume i. *Bull. Amer. Math. Soc*, *83*, 94–97.

Orchard, M., Kacprzynski, G., Goebel, K., Saha, B., & Vachtsevanos, G. (2008, oct.). Advances in uncertainty representation and management for particle filtering applied to prognostics. In *Prognostics and health management, 2008. phm 2008. international conference on* (p. 1 -6). doi: 10.1109/PHM.2008.4711433

Roychoudhury, I., Saxena, A., Celaya, J. R., & Goebel, K. (2013). Distilling the verification process for prognostics algorithms. In *2013 annual conference of the prognostics and health management society.*

Saha, B., Goebel, K., Poll, S., & Christophersen, J. (2009, feb.). Prognostics methods for battery health monitoring using a bayesian framework. *IEEE Transactions on Instrumentation and Measurement*, *58*(2), 291 -296. doi: 10.1109/TIM.2008.2005965

Sankararaman, S., Daigle, M., & Goebel, K. (2014, June). Uncertainty quantification in remaining useful life prediction using first-order reliability methods. *Reliability, IEEE Transactions on*, *63*(2), 603-619. doi: 10.1109/TR.2014.2313801

Sankararaman, S., & Goebel, K. (2013a). Remaining useful life estimation in prognosis: An uncertainty propagation problem. In *2013 aiaa infotech@ aerospace conference.*

Sankararaman, S., & Goebel, K. (2013b). Why is the remaining useful life prediction uncertain? In *Annual conference of the prognostics and health management society.*

Saxena, A., Celaya, J., Balaban, E., Goebel, K., Saha, B., Saha, S., & Schwabacher, M. (2008). Metrics for evaluating performance of prognostic techniques. In *Prognostics and health management, 2008. phm 2008. international conference on* (pp. 1–17).

Saxena, A., Celaya, J., Saha, B., Saha, S., & Goebel, K. (2010). Metrics for offline evaluation of prognostic performance. *International Journal of Prognostics and Health Management*, *1*(1), 20.

Saxena, A., Sankararaman, S., & Goebel, K. (2014). Performance evaluation for fleet-based and unit-based prognostic methods. In *Second european conference of the prognostics and health management society.*

Szabó, L. (2007). Objective probability-like things with and without objective indeterminism. *Studies In History and Philosophy of Science Part B: Studies In History and Philosophy of Modern Physics*, *38*(3), 626–634.

Wang, H.-F. (2011, January). Decision of prognostics and health management under uncertainty. *International Journal of Computer Applications*, *13*(4), 1–5. (Published by Foundation of Computer Science)

BIOGRAPHIES

Shankar Sankararaman received his B.S. degree in Civil Engineering from the Indian Institute of Technology, Madras in India in 2007 and later, obtained his Ph.D. in Civil Engineering from Vanderbilt University, Nashville, Tennessee, U.S.A. in 2012. His research focuses on the various aspects of uncertainty quantification, integration, and management in different types of aerospace, mechanical, and civil engineering systems. His research interests include probabilistic methods, risk and reliability analysis, Bayesian networks,

system health monitoring, diagnosis and prognosis, decision-making under uncertainty, treatment of epistemic uncertainty, and multidisciplinary analysis. He is a member of the Non-Deterministic Approaches (NDA) technical committee at the American Institute of Aeronautics, the Probabilistic Methods Technical Committee (PMC) at the American Society of Civil Engineers (ASCE), and the Prognostics and Health Management (PHM) Society. Currently, Shankar is a researcher at NASA Ames Research Center, Moffett Field, CA, where he develops algorithms for uncertainty assessment and management in the context of system health monitoring, prognostics, and decision-making.

Abhinav Saxena is a Research Scientist with SGT Inc. at the Prognostics Center of Excellence of NASA Ames Research Center, Moffett Field CA. His research focus lies in developing and evaluating prognostic algorithms for engineering systems using soft computing techniques. He has co-authored more than seventy technical papers including several book chapters on topics related to PHM. He is also a member of the SAE's HM-1 committee on Integrated Vehicle Health Management Systems and IEEE working group for standards on prognostics. Dr. Saxena is the editor-in-chief of International Journal of PHM and has led technical program committees in several PHM conferences. He is also a SGT technical fellow for prognostics. He has a PhD in Electrical and Computer Engineering from Georgia Institute of Technology, Atlanta. He earned his B.Tech in 2001 from Indian Institute of Technology (IIT) Delhi, and Masters Degree in 2003 from Georgia Tech. He has been a GM manufacturing scholar and is also a member of several professional societies for PHM including PHM Society, SAE, IEEE, AIAA, and ASME.

Kai Goebel is the Area Lead for Discovery and Systems Health at NASA Ames where he also directs the Prognostics Center of Excellence. After receiving the Ph.D. from the University of California at Berkeley in 1996, Dr. Goebel worked at General Electric's Corporate Research Center in Niskayuna, NY from 1997 to 2006 as a senior research scientist before joining NASA. He has carried out applied research in the areas of artificial intelligence, soft computing, and information fusion and his interest lies in advancing these techniques for real time monitoring, diagnostics, and prognostics. He holds 18 patents and has published more than 300 papers in the area of systems health management.